Examen ING3-GEE EILCO - Intelligence Artificielle
Rattrapage

Printemps 2024

Nom :
Prénom :

Total: 27 points
Durée: 2h
Instructions générales: L'examen comprend 2 parties (Chacune de ces parties reprenant différentes sous-questions). Vous êtes libres de rédiger vos réponses sur des pages supplémentaires en veillant toutefois à bien indiquer le numéro de chaque question. Une fois l'examen terminé, Assurez vous de bien écrire votre nom (de façon lisible) sur chacune des pages. Répondez à un maximum de questions, en commençant par les questions qui vous semblent les plus abordables.

Partie 1 (12pts)

1. [5pts] *Indiquer si les affirmations suivantes sont vraies ou fausses*

Vrai / Faux	*Le phénomène d'overfitting se produit lorsqu'un modèle est efficace sur les données d'entraînement mais retourne des prédictions incohérentes sur de nouvelles données qui n'étaient pas présentes dans l'ensemble d'entrainement*
Vrai / Faux	*Soit $\boldsymbol{X}$ une matrice de taille $n \times p$ et $\boldsymbol{t}$ un vecteur de taille n. Le vecteur de coefficients $\boldsymbol{\beta}$ estimé en minimisant la somme des carrés des résidus est donné par $\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{t}$.*
Vrai / Faux	*La fonction de coût utilisée en régression logistique est la somme des carrés des résidus, tout comme pour la régression linéaire*
Vrai / Faux	*La régression linéaire par sélection du meilleur sous-ensemble pour un vecteur $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_p] \in \mathbb{R}^{p+1}$ requiert d'apprendre $\binom{n}{p+1}$ modèles*
Vrai / Faux	*La régression linéaire par sélection du meilleur sous-ensemble pour un vecteur $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_p] \in \mathbb{R}^{p+1}$ requiert d'apprendre $\binom{n}{p}$ modèles*
Vrai / Faux	*Dans un modèle de régression linéaire centré. I.e. tel que les caractéristiques satisfont $\sum_{i=1}^n X_j^{(i)} = 0$ pour tout j, et pour une fonction de coût donnée par $\frac{1}{2N}\sum_{i=1}^N (t^{(i)} - \beta_0 - \sum_{j=1}^d \beta_j X_j^{(i)})^2$ on a nécessairement $\beta_0 = \frac{1}{N}\sum_{i=1}^N t^{(i)}$*
Vrai / Faux	*Dans un modèle de type Ridge centré. I.e. tel que les caractéristiques satisfont $\sum_{i=1}^n X_j^{(i)} = 0$ pour tout j, et pour une fonction de coût donnée par $\frac{1}{2N}\sum_{i=1}^N (t^{(i)} - \beta_0 - \sum_{j=1}^d \beta_j X_j^{(i)})^2$ on a nécessairement $\beta_0 = \frac{1}{N}\sum_{i=1}^N t^{(i)}$*
Vrai / Faux	*L'erreur quadratique moyenne pour une famille de modèles peut sécrire comme la somme du carré du biais et de la variance*
Vrai / Faux	*Un modèle de classification linéaire entrainé par minimisation de la somme des carrés*

	des résidus sur un jeu de données linéairement séparables ne violera jamais la contrainte de séparation des classes
Vrai / Faux	*La régression Ridge peut s'appliquer à des données dont la matrice caractéristique est singulière ou mal conditionnée alors que la régression linéaire simple ne peut pas*

2. [4pts] *On considère le réseau de neurones représenté à la figure 4*

   (a) [2pts] *On commence par supposer que toutes les fonctions d'activation sont des fonctions Relu*

   $$\sigma(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{1}$$

   et que tous les biais sont unitaires ($w_{i0}^{(\ell)} = 1$ pour tout neurone) En supposant $x_1 = -1$ et $x_2 = 1$, donner la sortie des trois neurones de la couche 4.

   (b) [2pts] *On suppose à présent que les fonctions d'activation sont des sigmoïdes. Soit δ_1^4, δ_2^4 et δ_3^4 les dérivées $\frac{\partial L}{\partial a_i^4}$ de la fonction de coût par rapport aux valeurs de préactivation des trois neurones de la couche 4 et soit δ_1^3 la dérivée par rapport à la valeur de préactivation du neurone de la couche 3, $\frac{\partial L}{\partial a_1^3}$. Donner la relation qui permet de dériver δ_1^3 à partir de δ_1^4, δ_2^4 et δ_3^4.*

3. [3pts] *Soit $\boldsymbol{X} \in \mathbb{R}^{n\times 2}$ une matrice de caractéristiques à laquelle est associé le vecteur cible $\boldsymbol{t} \in \mathbb{R}^n$. On suppose que la matrice $\boldsymbol{X}^T\boldsymbol{X}$ est inversible et que les données sont centrées ($\sum_{i=1}^n t^{(i)} = \sum_{i=1}^n x_j^{(i)} = 0$) et on souhaite déterminer le vecteur $\boldsymbol{\beta}$ qui minimise la fonction de coût*

   $$\frac{1}{n}\sum_{i=1}^{n}\left(t^{(i)} - (\beta_1 x_1^{(i)} + \beta_2 x_2^{(i)})\right)^2 + \lambda\sum_{j=1}^{2} w_j\beta_j^2$$

   Les $w_j > 0$ sont des poids positifs qui accordent plus ou moins d'importance aux coefficients β_j. On notera que la pénalité peut se réécrire $\lambda\sum_{j=1}^2 w_j\beta_j^2 = \boldsymbol{\beta}^T\boldsymbol{W}\boldsymbol{\beta}$ où la matrice $\boldsymbol{W}$ est la matrice diagonale donnée par $\boldsymbol{W} = \begin{bmatrix} w_1 & 0 \\ 0 & w_2 \end{bmatrix}$ et $\boldsymbol{\beta} = [\beta_1, \beta_2]$. Dériver (en donnant toutes les étapes) les équations normales ainsi que l'expression de $\boldsymbol{\beta}$.

Partie 2 (15pts)

1. [5pts] *Indiquer si les affirmations suivantes sont vraies ou fausses*

Vrai / Faux	*Le gradient de la fonction d'entropie binaire croisée pour le modèle de régression logistique et pour une unique observation est donné par* $\text{grad}_{\boldsymbol{\beta}}L = (1 - \sigma(\beta_0 + \sum_{j=1}^{d} \beta_j X_j))\tilde{\boldsymbol{X}}$ *où* $\tilde{\boldsymbol{X}} = [1, X_1, \ldots, X_d]$ *représente le vecteur caractéristique.*
Vrai / Faux	*Dans la backpropagation, le gradient de la fonction de coût L par rapport à la valeur de préactivation* a_i *de la couche* ℓ *est donné par* $\delta_i^{(\ell)} \equiv \frac{\partial L}{\partial a_i^{\ell}} = \sum_{j\in(\ell-1)} \delta_j^{(\ell-1)} w_{ji}^{(\ell-1)} \sigma'(a_i^{(\ell)})$ *où* $\delta_j^{(\ell-1)} = \frac{\partial L}{\partial a_j^{\ell-1}}$ *est le gradient de la fonction de coût par rapport à la valeur de préactivation de la couche* $(\ell - 1)$
Vrai / Faux	*En régression linéaire, pour des données centrées telles que* $\sum_{i=1}^{n} X_j^{(i)} = \sum_{i=1}^{n} t^{(i)} = 0$ *ajouter une pénalité de type Ridge d'importance croissante va avoir pour effet de faire tendre le modèle vers le plan* $t = 0$
Vrai / Faux	*On considère un jeu de données constitué de deux classes concentriques et distribuées sur deux anneaux disjoints. Un réseau de neurones dont la première couche est constituée de deux neurones et la couche de sortie ne contient qu'un seul neurone sera incapable de séparer les deux classes*
Vrai / Faux	*Pour qu'un réseau de neurones puisse être utilisé dans le cadre d'un problème de classification, sa dernière couche doit être constituée d'au moins 2 neurones*
Vrai / Faux	*Un réseau de neurones à une couche dont tous les neurones sont munis d'une fonction d'activation linéaire est équivalent à un modèle de régression linéaire*
Vrai / Faux	*Dans un réseau de neurones convolutif, un max pooling avec une fenêtre de taille* 2×2 *et un stride de* 2 *réduit la résolution de moitié dans les deux dimensions spatiales.*

2. [4pts] *Dans cette question, on s'intéresse aux deux modèles de régularisation Ridge et Lasso.*

   (a) [2pts] *Donner, pour chacun des modèles, l'expression de la fonction de coût associée.*

   (b) [2pts] *On suppose que l'on utilise chacune des approches (en considérant un paramètre de régularisation suffisamment élevé) pour entrainer un modèle de degré 20 sur des données distribuées approximativement de manière quadratique. Les coefficients des deux modèles sont représentés à la figure 3 (gauche et droite). Indiquer sur chaque figure le modèle qui lui correspond.*

3. [6pts] *Vous disposez d'un extrait de code donné à la figure 1. On souhaite compléter cet extrait de façon à entrainer un modèle linéaire sur les données fournies à la figure 2 (haut) à l'aide dune descente de gradient.*

   (a) [2pts] *On commencera par générer des caractéristique de degré maximum égal à deux. Donner dans l'espace ci-dessous, les lignes à ajouter à l'Extrait 1 (figure 1) afin de générer ces caractéristiques* *(b) [4pts] On souhaite ensuite implémenter les itérations de descente de gradient. Compléter la boucle while en calculant le gradient et en mettant à jour le vecteur des coefficients de régression. On rapportera les lignes à ajouter dans le rectangle ci-dessous.*

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures


x = np.linspace(-2,2,10)

'''# (1) generation des caracteristiques '''



'''# (2) descente de gradient '''

maxIter = 100
current_iter = 0
loss = np.zeros((maxIter, 1))
beta = np.random.normal(0,1,D+1)

eta= .01


while current_iter<maxIter:

    '''Completez votre reponse en calculant le gradient et en mettant
    a jour le vecteur des coefficients de regression ici'''

```

Figure 1: Extrait 1

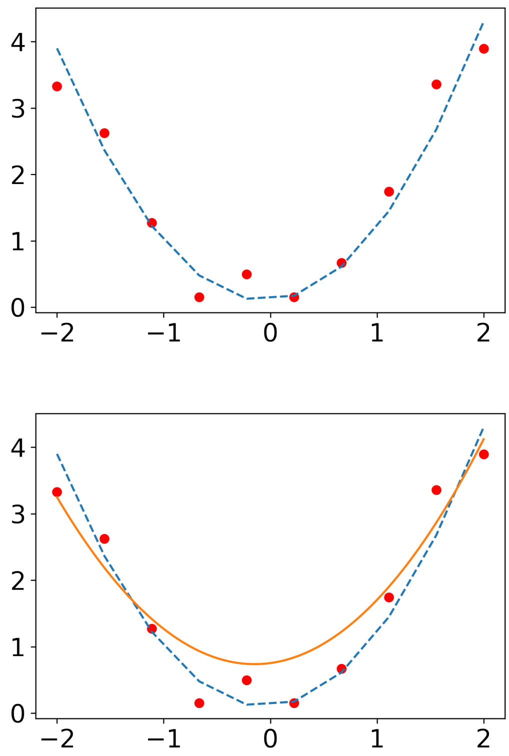

Figure 2: Jeu de données et résultat souhaité pour la question 2.3.

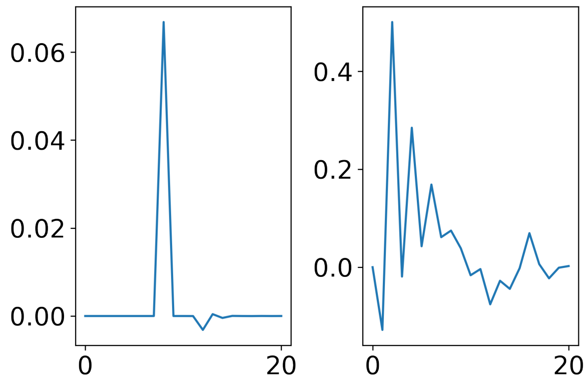

Figure 3: Représentation des coefficients de régression pour la question 2.2

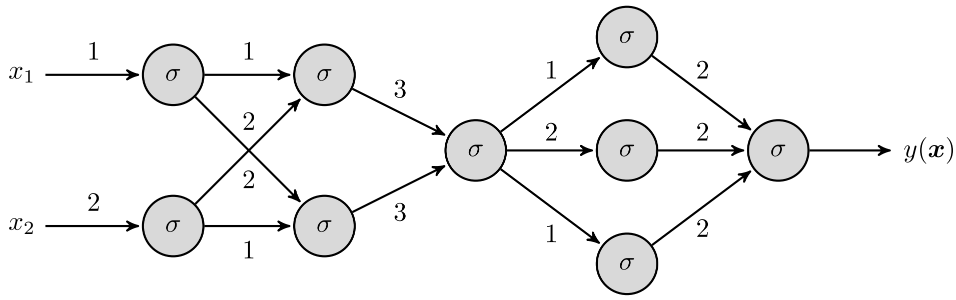

Figure 4: Réseau de neurones utilisé pour la question 1.2